WEST Search History

DATE: Tuesday, October 14, 2003

Set Name side by side	Query	Hit Count	Set Name result set
DB=JPAB,EPAB,DWPI,TDBD; PLUR=YES; OP=ADJ			
L10	L9 and compil$	2	L10
L9	instrument$ and stub	110	L9
DB=USPT,PGPB; PLUR=YES; OP=ADJ			
L8	L7 and L1	29	L8
L7	(((717/124 |717/125 |717/126 |717/127 |717/128 |717/129 |717/130 |717/131 |717/132 |717/133)!.CCLS.))	1107	L7
DB=USPT; PLUR=YES; OP=ADJ			
L6	L5 and GUI	37	L6
L5	L2 and compil$	70	L5
L4	L2 and GUI	38	L4
L3	L2	117	L3
DB=USPT,PGPB; PLUR=YES; OP=ADJ			
L2	L1 and pars$	172	L2
L1	instrument$ and stub	3699	L1

END OF SEARCH HISTORY

Welcome to IEEE Xplore

- Home
- What Can I Access?
- Log-out

Tables of Contents

- Journals & Magazines
- Conference Proceedings
- Standards

Search

- By Author
- Basic
- Advanced

Member Services

- Join IEEE
- Establish IEEE Web Account
- Access the IEEE Member Digital Library

Print Format

Your search matched **4** of **976857** documents.

A maximum of **4** results are displayed, **15** to a page, sorted by **Relevance** in **descending** order.
You may refine your search by editing the current search expression or entering a new one the text box.

Then click **Search Again**.

(generating stub)

Search Again

Results:
Journal or Magazine = **JNL** Conference = **CNF** Standard = **STD**

1 **Techniques for optimizing CORBA middleware for distributed embedded systems**
Gokhale, A.; Schmidt, D.C.;
INFOCOM '99. Eighteenth Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings. IEEE , Volume: 2 , 21-25 March 1999
Page(s): 513 -521 vol.2

[Abstract] [PDF Full-Text (824 KB)] **IEEE CNF**

2 **MysterX: a Scheme toolkit for building interactive applications with COM**
Steckler, P.A.;
Technology of Object-Oriented Languages and Systems, 1999. TOOLS 30. Proceedings , 1-5 Aug. 1999
Page(s): 364 -373

[Abstract] [PDF Full-Text (164 KB)] **IEEE CNF**

3 **Mockingbird: flexible stub compilation from pairs of declarations**
Auerbach, J.; Barton, C.; Chu-Carroll, M.; Raghavachari, M.;
Distributed Computing Systems, 1999. Proceedings. 19th IEEE International Conference on , 31 May-4 June 1999
Page(s): 393 -402

[Abstract] [PDF Full-Text (104 KB)] **IEEE CNF**

4 **Improving driver robustness: an evaluation of the Devil approach**

Reveillere, L.; Muller, G.;

Dependable Systems and Networks, 2001. Proeedings. The International Conference on , 1-4 July 2001

Page(s): 131 -140

[Abstract] [PDF Full-Text (744 KB)] **IEEE CNF**

Search Results for: **[instrument<AND>((generating stub))]**
Found **17** of **121,350 searched.**

Search within Results

> Advanced Search
> Search Help/Tips

Sort by: Title Publication Publication Date Score Binder

Results 1 - 17 of 17 short listing

1 Technical papers: software design: DADO: enhancing middleware to support crosscutting features in distributed, heterogeneous systems 84%
Eric Wohlstadter , Stoney Jackson , Premkumar Devanbu
Some "non-" or "extra-functional" features, such as reliability, security, and tracing, defy modularization mechanisms in programming languages. This makes such features hard to design, implement, and maintain. Implementing such features within a single platform, using a single language, is hard enough. With distributed, heterogeneous (DH) systems, these features induce complex implementations which cross-cut different languages, OSs, and hardware platforms, while still needing to share data and ...

2 Evolving RPC for active storage 80%
Muthian Sivathanu , Andrea C. Arpaci-Dusseau , Remzi H. Arpaci-Dusseau
Tenth international conference on architectural support for programming languages and operating systems on Proceedings of the 10th international conference on architectural support for programming languages and operating systems (ASPLOS-X) October 2002
Volume 37 , 36 , 30 Issue 10 , 5 , 5
We introduce Scriptable RPC (SRPC), an RPC-based framework that enables distributed system services to take advantage of active components. Technology trends point to a world where each component in a system (whether disk, network interface, or memory) has substantial computational capabilities; however, traditional methods of building distributed services are not designed to take advantage of these new architectures, mandating wholesale change of the software base to exploit more powerful hardw ...

3 Lightweight remote procedure call 77%
Brian N. Bershad , Thomas E. Anderson , Edward D. Lazowska , Henry M. Levy
ACM Transactions on Computer Systems (TOCS) February 1990
Volume 8 Issue 1
Lightweight Remote Procedure Call (LRPC) is a communication facility designed and optimized

for communication between protection domains on the same machine. In contemporary small-kernel operating systems, existing RPC systems incur an unnecessarily high cost when used for the type of communication that predominates—between protection domains on the same machine. This cost leads system designers to coalesce weakly related subsystems into the same protection domain, trading safety for ...

4 Lightweight remote procedure call 77%

B. Bershad , T. Anderson , E. Lazowska , H. Levy

ACM SIGOPS Operating Systems Review , Proceedings of the twelfth ACM symposium on Operating systems principles November 1989

Volume 23 Issue 5

Lightweight Remote Procedure Call (LRPC) is a communication facility designed and optimized for communication between protection domains on the same machine. In contemporary small-kernel operating systems, existing RPC systems incur an unnecessarily high cost when used for the type of communication that predominates — between protection domains on the same machine. This cost leads system designers to coalesce weakly-related subsystems into the same protection domain, tradin ...

5 Performance and scalability of EJB applications 77%

Emmanuel Cecchet , Julie Marguerite , Willy Zwaenepoel

ACM SIGPLAN Notices , Proceedings of the 17th ACM conference on Object-oriented programming, systems, languages, and applications November 2002

Volume 37 Issue 11

We investigate the combined effect of application implementation method, container design, and efficiency of communication layers on the performance scalability of J2EE application servers by detailed measurement and profiling of an auction site server.We have implemented five versions of the auction site. The first version uses stateless session beans, making only minimal use of the services provided by the Enterprise JavaBeans (EJB) container. Two versions use entity beans, one with container- ...

6 Recompilation for debugging support in a JIT-compiler 77%

Mustafa M. Tikir , Jeffrey K. Hollingsworth , Guei-Yuan Lueh

ACM SIGSOFT Software Engineering Notes , Proceedings of the 2002 ACM SIGPLAN-SIGSOFT workshop on Program analysis for software tools and engineering November 2002

Volume 28 Issue 1

A static Java compiler converts Java source code into a verifiably secure and compact architecture-neutral intermediate format, called Java *byte codes*. The Java byte codes can be either interpreted by a Java Virtual Machine or translated into native code by Java Just-In-Time compilers. Static Java compilers embed debug information in the Java class files to be used by the source level debuggers. However, the debug information is generated for architecture independent byte codes and most o ...

7 Profile-guided code compression 77%

Saumya Debray , William Evans

ACM SIGPLAN Notices , Proceedings of the ACM SIGPLAN 2002 Conference on Programming language design and implementation May 2002

Volume 37 Issue 5

As computers are increasingly used in contexts where the amount of available memory is limited, it becomes important to devise techniques that reduce the memory footprint of application programs while leaving them in an executable form. This paper describes an approach to applying data compression techniques to reduce the size of infrequently executed portions of a program. The compressed code is decompressed dynamically (via software) if needed, prior to execution. The use of data compression t ...

8 Evaluating the performance limitations of MPMD communication 77%

Chi-Chao Chang , Grzegorz Czajkowski , Thorsten von Eicken , Carl Kesselman

Proceedings of the 1997 ACM/IEEE conference on Supercomputing (CDROM) November 1997

The MPMD approach for parallel computing is attractive for programmers who seek fast development cycles, high code re-use, and modular programming, or whose applications exhibit irregular computation loads and communication patterns. RPC is widely adopted as the communication abstraction for crossing address space boundaries. However, the communication overheads of existing RPC-based systems are usually an order of magnitude higher than those found in highly tuned SPMD systems. This problem has ...

9 Efficient wire formats for high performance computing 77%

Fabian E. Bustamante , Greg Eisenhauer , Karsten Schwan , Patrick Widener

Proceedings of the 2000 ACM/IEEE conference on Supercomputing (CDROM) November 2000

High performance computing is being increasingly utilized in non-traditional circumstances where it must interoperate with other applications. For example, online visualization is being used to monitor the progress of applications, and real-world sensors are used as inputs to simulations. Whenever these situations arise, there is a question of what communications infrastructure should be used to link the different components. Traditional HPC-style communications systems such as MPI offer r ...

10 SAFKASI: a security mechanism for language-based systems 77%

Dan S. Wallach , Andrew W. Appel , Edward W. Felten

ACM Transactions on Software Engineering and Methodology (TOSEM) October 2000 Volume 9 Issue 4

In order to run untrusted code in the same process as trusted code, there must be a mechanism to allow dangerous calls to determine if their caller is authorized to exercise the privilege of using the dangerous routine. Java systems have adopted a technique called stack inspection to address this concern. But its original definition, in terms of searching stack frames, had an unclear relationship to the actual achievement of security, overconstrained the implementation of a Java system, lim ...

11 Practicing JUDO: Java under dynamic optimizations 77%

Micha? Cierniak , Guei-Yuan Lueh , James M. Stichnoth

ACM SIGPLAN Notices , Proceedings of the ACM SIGPLAN 2000 conference on Programming language design and implementation May 2000 Volume 35 Issue 5

A high-performance implementation of a Java Virtual Machine (JVM) consists of efficient implementation of Just-In-Time (JIT) compilation, exception handling, synchronization mechanism, and garbage collection (GC). These components are tightly coupled to achieve high

performance. In this paper, we present some static anddynamic techniques implemented in the JIT compilation and exception handling of the Microprocessor Research Lab Virtual Machine (MRL VM), ...

12 The runtime creation of code for printing simulation output 77%
John H. Reynolds
Proceedings of the 22nd conference on Winter simulation December 1990

13 Hybrid domain-specific kits for a flexible software factory 77%
Martin L. Griss , Kevin D. Wentzel
Proceedings of the 1994 ACM symposium on Applied computing April 1994

14 An object-based infrastructure for program monitoring and steering 77%
Greg Eisenhauer , Karsten Schwan
Proceedings of the SIGMETRICS symposium on Parallel and distributed tools August 1998

15 Flick: a flexible, optimizing IDL compiler 77%
Eric Eide , Kevin Frei , Bryan Ford , Jay Lepreau , Gary Lindstrom
ACM SIGPLAN Notices , Proceedings of the ACM SIGPLAN 1997 conference on Programming language design and implementation May 1997
Volume 32 Issue 5
An interface definition language (IDL) is a nontraditional language for describing interfaces between software components. IDL compilers generate "stubs" that provide separate communicating processes with the abstraction of local object invocation or procedure call. High-quality stub generation is essential for applications to benefit from component-based designs, whether the components reside on a single computer or on multiple networked hosts. Typical IDL compilers, ...

16 Separating data and control transfer in distributed operating systems 77%
Chandramohan A. Thekkath , Henry M. Levy , Edward D. Lazowska
Proceedings of the sixth international conference on Architectural support for programming languages and operating systems November 1994
Volume 29 , 28 Issue 11 , 5
Advances in processor architecture and technology have resulted in workstations in the 100+ MIPS range. As well, newer local-area networks such as ATM promise a ten- to hundred-fold increase in throughput, much reduced latency, greater scalability, and greatly increased reliability, when compared to current LANs such as Ethernet.We believe that these new network and processor technologies will permit tighter coupling of distributed systems at the hardware level, and that distribu ...

17 The POLYLITH software bus 77%
James M. Purtilo
ACM Transactions on Programming Languages and Systems (TOPLAS) January 1994
Volume 16 Issue 1
We describe a system called POLYLITH that helps programmers prepare and interconnect mixed-language software components for execution in heterogeneous environments.

POLYLITH's principal benefit is that programmers are free to implement functional requirements separately from their treatment of interfacing requirements; this means that once an application has been developed for use in one execution environment (such as a distributed network) it can be adapted for reuse in other environments ...

Results 1 - 17 of 17 short listing
